Attorney's Docket No. RAB 97-002 ***PATENT***

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Anticipated Classification of this application:
Class 395 Subclass 416, 403, 417
Prior application:
Examiner: Nguyen, T
Art Unit: 2309

Box Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

TRANSMITTAL OF FILING UNDER 37 C.F.R. 1.60(b)

WARNING: *A C-I-P (continuation-in-part)* ***cannot*** *be filed under 37 CFR 1.60(b).*

WARNING: *A filing under 37 C.F.R. § 1.60(b) can only be made if the "prior application was a nonprovisional application and a complete application as set forth in § 1.51(a)(1)." 37 C.F.R. § 1.60(b)(1).*

WARNING: *Filing under 37 CFR 1.60 is permitted only if filed by the same or less than all the inventors named in the prior application. 37 CFR 1.60(b)(3).*

WARNING: *The filing of an application at the United States stage of an International Application requires an oath or declaration. 37 CFR 1.61(a)(4).*

WARNING: *The claims of this new application may be finally rejected in the first Office action where all claims of the new application are drawn to the same invention claimed in the earlier application and would have been properly finally rejected on the grounds or art of record in the next Office action if they had been entered in the earlier application. MPEP § 706.07(b).*

This is a request for filing a

☒ Continuation

☐ Divisional

application under 37 CFR 1.60, of pending prior application
Serial No. 08/458,479 filed on 6/2/95
Date

CERTIFICATION UNDER 37 CFR 1.10

I hereby certify that this 37 CFR 1.60 request and the documents referred to as attached therein are being deposited with the United States Postal Service on this date August 4, 1997 in an envelope as "Express Mail Post Office to Addressee" service under 37 CFR 1.10, Mailing Label Number EG459773316US addressed to the: Assistant Commissioner for Patents, Washington, D.C. 20231.

J. Nicholas Gross
(type or print name of person mailing paper)

J. Nicholas Gross
Signature of person mailing paper

NOTE: *Each paper or fee filed by "Express Mail" must have the number of the "Express Mail" mailing label placed thereon prior to mailing. (37 CFR 1.10(b)).*

WARNING: *Certificate of mailing (first class) or facsimile transmission procedures of 37 CFR 1.8 cannot be used to obtain a date of mailing or transmission for this correspondence.*

RAB97-002

of Richard A. Belgard
Inventor(s)

for Address Translation Method and Mechanism Using Physical Address Information During a Segmentation Process
Title of Invention

NOTE: 37 CFR 1.60 permits the omission of a declaration only if the prior application was complete as set forth in 37 CFR 1.51(a), namely, the prior application comprised at least (1) a specification, including a claim or claims; (2) a declaration; (3) drawings when necessary; and (4) the prescribed filing fee. Accordingly, as presently worded, 37 CFR 1.60 does not permit this procedure to be used where the prior application is pending but only the processing and retention fee required by 37 CFR 1.21(l) is paid or where the declaration was not filed.

1. Copy of Prior Application as Filed That is Attached

NOTE: Under 37 CFR 1.60, practice signing and execution of the application by the applicant may be omitted provided the copy is supplied by and accompanied by a statement by the applicant or his or her attorney or agent that the application papers comprise a true copy of the prior application as filed and that no amendments referred to in the declaration filed to complete the prior application introduced new matter therein.

NOTE: This statement need not be verified if made by an attorney registered to practice before the PTO. (37 CFR 1.60(b)).

☒ I hereby verify that the attached papers are a true copy of what is shown in my records to be the above identified prior application, including the oath or declaration originally filed. (37 C.F.R. 1.60(b)(2))

The copy of the papers of prior application as filed which are attached are as follows:

☒ 21 page(s) of specification
☒ 9 page(s) of claims
☒ 1 page(s) of abstract
☒ 5 sheet(s) of drawing

(also complete part 6 below, if drawings are to be transferred)

☒ 1 pages of declaration and power of attorney

(If the copy of the declaration being filed does not show applicant's signature, because the attorney's records do not contain a copy of the signed declaration actually filed for the application, indicate thereon that it was signed and complete the following:)

☐ In accordance with the indication required by 37 C.F.R. 60(b), my records reflect that the original signed declaration showing applicant's signature was filed on _____.

☐ The amendment referred to in the declaration filed to complete the prior application and I hereby state, in accordance with the requirements of 37 CFR 1.60(b), that this amendment did not introduce new matter therein.

RAB97-002

2. Amendments

WARNING: *"The claim of a new application may be finally rejected in the first Office action in those situations where (1) the new application is a continuing application of, or a substitute for, an earlier application, and (2) all the claims of the new application (a) are drawn to the same invention claimed in the earlier application, and (b) would have been properly finally rejected on the grounds or art of record in the next Office action if they had been entered in the earlier application." MPEP § 706.07(b).*

☒ Cancel in this application original claims 6-37 of the prior application before calculating the filing fee. (At least one original independent claim must be retained for filing purposes.)

☒ A preliminary amendment is enclosed. (Claims added by this amendment have been properly numbered consecutively beginning with the number next following the highest numbered original claim in the prior application.)

NOTE: Only amendments reducing the number of claims or adding a reference to the prior application (§ 1.78(a)) will be entered before calculating the filing fee and granting the filing date. 37 CFR 1.60(b)(4).

NOTE: "When filing under Rule 1.60 retain at least one original claim from the patent application to assure a complete application." Notice of March 3, 1986 (1064 O.G. 37-38).

3. Petition for Suspension of Prosecution for the Time Necessary to File an Amendment

NOTE: Where it is possible that the claims on file will give rise to a first action final for this continuation application and for some reason an amendment cannot be filed promptly (e.g., experimental data is being gathered) it may be desirable to file a petition for suspension of prosecution for the time necessary).

(check the next item, if applicable)

☐ There is provided herewith a Petition To Suspend Prosecution For The Time Necessary to File An Amendment (New Application Filed Concurrently).

4. Information Disclosure Statement

(check this item, if applicable)

☒ An information disclosure statement is submitted herewith.

RAB 97-002

5. Fee Calculation (37 CFR 1.16)

CLAIMS AS FILED					
Number filed		Number Extra		Rate	Basic Fee 37 CFR 1.16(a) $770.00
Total Claims (37 CFR 1.16(c))	48	−20= 28	×	$ 22.00	$616.00
Independent Claims (37 CFR 1.16(b))	11	−3= 8	×	$ 80.00	$640.00
Multiple dependent claim(s), if any (37 CFR 1.16(d))			+	$260.00	

☐ Fee for extra claims is not being paid at this time. (37 C.F.R 1.16(d))

NOTE: If the fees for extra claims are not paid on filing they must be paid or the claims cancelled by amendment, prior to the expiration of the time period set for response by the PTO in any notice of fee deficiency. 37 CFR 1.16(d).

Filing Fee Calculation $2026.00

6. Small Entity Status

☒ A verified statement that this filing is by a small entity:

☐ is attached.

☒ has been filed in the parent application and such status is still proper and desired. (37 CFR 1.28(a))

Filing Fee Calculation (50% of above) $1013.00

NOTE: Any excess of the full fee paid will be refunded if a verified statement is filed within 2 months of the date of timely payment of a full fee then the excess fee paid will be refunded on request. 37 CFR 1.28(a).

NOTE: 37 CFR 1.28(a), last sentence states: "Applications filed under § 1.60 or § 1.62 of this part must include a reference to a verified statement in a parent application if status as a small entity is still proper and desired."

7. Drawings

☒ Drawings are enclosed

☐ Formal

☒ Informal

WARNING: *DO NOT submit original drawings. A high quality copy of drawings should be supplied when filing a patent application. The drawings that are submitted to the Office must be on strong, white, smooth, and non-shiny paper and meet the standards of § 1.84. If corrections to the drawings are necessary, they should be made to the original drawings and a high-quality copy of the corrected original drawing then submitted to the Office. Only one copy is required or desired. Comments on proposed new 37 CFR 1.84. Notice of March 9, 1988 (1090 O.G. 57-62).*

NOTE: "Identifying indicia, if provided, should include the application number or the title of the invention, inventor's name, docket number (if any), and the name and telephone number of a person to call if the Office is unable to match the drawings to the proper application. This information should be placed on the back of each sheet of drawing a minimum distance of 1.5 cm. (5/8 inch) down from the top of the page." 37 C.F.R. 1.84(c)).

RAB97-002

8. Priority—35 U.S.C. 119

☐ Priority of application Serial No. 0 /_____ filed on _____ in _____ is claimed under 35 U.S.C. 119. Country

☐ The certified copy has been filed in prior U.S. application Serial No. 0 /_____ on _____.

☐ The certified copy will follow.

9. Relate Back—35 U.S.C. 120

☒ Amend the specification by inserting, before the first line, the following sentence: "This is a

☒ continuation

☐ divisional

of copending application(s)

☒ Serial number 08/ 458,479 filed on June 2, 1995 "

☐ International Application _____ filed on _____ that designated the U.S."

NOTE: The proper reference to a prior filed PCT application that entered the U.S. national phase is the U.S. serial number and the filing date of the PCT application which designated the U.S.

10. Inventorship Statement

NOTE: "If the continuation or divisional application is filed by less than all the inventors named in the prior application a statement must accompany the application when filed requesting deletion of the names of the person or persons who are not inventors of the invention being claimed in the continuation or divisional application." 37 CFR 1.60(b)(4) [emphasis added].

(complete appropriate items (a) and (b))

(a) With respect to the prior copending U.S. application from which this application claims benefit under 35 U.S.C. 120, the inventor(s) in this application is (are):

(complete applicable item below)

☒ the same.

☐ less than those named in the prior application. It is requested that the following inventor(s) identified above for the prior application be deleted:

_____

(type name(s) of inventor(s) to be deleted)

(b) The inventorship for all the claims in this application are

☒ the same.

☐ not the same. And an explanation, including the ownership of the various claims at the time the last claimed invention was made, is submitted.

RAB97-002

11. Assignment

☐ The prior application is assigned of record to

_____ .

☐ An assignment of the invention to _____

_____

is attached. A separate ☐ "COVER SHEET FOR ASSIGNMENT (DOCUMENT) ACCOMPANYING NEW PATENT APPLICATION" or ☐ FORM PTO 1595 is also attached.

NOTE: "If an assignment is submitted with a new application, send two separate letters - one for the application and one for the assignment." Notice of May 4, 1990 (1114 O.G. 77-78).

NOTE: When an assignee files a . . . divisional application (under . . . 1.60 . . .) reference may be made to a statement filed under 37 CFR 3.73(b) in the parent application, or a copy of that statement may be filed Notice of April 30, 1993, 1150 O.G. 62-64.

12. Fee Payment Being Made At This Time

☐ Not Enclosed

☐ No filing fee is submitted.
(This and the surcharge required by 37 CFR 1.16(e) can be paid subsequently).

☒ Enclosed

☒ basic filing fee $ 1013

☐ recording assignment
($40.00; 37 CFR 1.21(h))
(See attached "COVER SHEET FOR ASSIGNMENT ACCOMPANYING NEW PATENT APPLICATION".)

☐ processing and retention fee
($130.00; 37 CFR 1.53(d) and 1.21(l)) $_____

NOTE: 37 CFR 1.21(l) establishes a fee for processing and retaining any application which is abandoned for failing to complete the application pursuant to 37 CFR 1.53(d) and this, as well as the changes to 37 CFR 1.53 and 1.78 indicate that in order to obtain the benefit of a prior U.S. application, either the basic filing fee must be paid or else the processing and retention fee of § 1.21(l) must be paid within 1 year from notification under § 53(d).

Total fees enclosed $_____

13. Method of Payment of Fees

☒ Enclosed is a check in the amount of $ 1013 .

☐ Charge Account No. _____ in the amount of $_____.
A duplicate of this request is attached.

NOTE: Fees should be itemized in such a manner that is clear for which purpose the fees are paid. 37 CFR 1.22(b).

RAB97-002

14. Authorization To Charge Additional Fees

WARNING: *If no fees are being paid on filing do not complete this item.*

WARNING: *Accurately count claims, especially multiple dependent claims, to avoid unexpected high charges if extra claim charges are authorized.*

☐ The Commissioner is hereby authorized to charge the following additional fees which may be required by this paper and during the entire pendency of the application to Account No. _____.

☐ 37 C.F.R. 1.16 (a), (f) or (g) (filing fees)

☐ 37 C.F.R. 1.16 (b), (c) and (d) (presentation of extra claims)

NOTE: Because additional fees for excess or multiple dependent claims not paid on filing or on later presentation must only be paid or these claims cancelled by amendment prior to the expiration of the time period set for response by the PTO in any notice of fee deficiency (37 CFR 1.16(d)) it might be best not to authorize the PTO to charge additional claim fees, except possibly when dealing with amendments after final action.

☐ 37 C.F.R. 1.17 (application processing fees)

WARNING: *While 37 CFR 1.17(a), (b), (c) and (d) deal with extensions of time under § 1.136(a) this authorization should be made only with the knowledge that: "Submission of the appropriate extension fee under 37 CFR 1.136(a) is to no avail unless a request or petition for extension is filed." [emphasis added]. Notice of November 5, 1985 (1060 O.G. 27).*

☐ 37 C.F.R. 1.18 (issue fee at or before mailing Notice of Allowance, pursuant to 37 CFR 1.311(b)).

NOTE: Where an authorization to charge the issue fee to a deposit account has been filed before the mailing of a Notice of Allowance, the issue fee will be automatically charged to the deposit account at the time of mailing the notice of allowance. 37 CFR 1.311(b)).

NOTE: 37 CFR 1.28(b) requires "Notification of any change in status resulting in loss of entitlement to small entity status must be filed in the application . . . prior to paying or at the time of paying . . . issue fee." From the wording of 37 CFR 1.28(b): (a) notification of change of status must be made even if the fee is paid as "other than a small entity" and (b) no notification is required if the change is to another small entity.

15. Power of Attorney

☒ The power of attorney in the prior application is to

Lawt; J. Nicholas Gross / Peter Courture 34175

Attorney Reg. No.

a. ☒ The power appears in the original papers in the prior application.

b. ☐ Because the power does not appear in the original papers, a copy of the power in the prior application is enclosed.

c. ☐ A new power has been executed and is attached.

d. ☒ Address all future communications to

(item d may only be completed by applicant, or attorney or agent of record)

Peter Courture
Lawt
993 Highland Circle
Los Altos, CA 94024

RAB97-002

16. Maintenance of Copendency of Prior Application

(this item must be completed and the papers filed in the prior application if the period set in the prior application has run)

☐ A petition, fee and response has been filed to extend the term in the pending prior application until _____ .

NOTE: *The PTO finds it useful if a copy of the petition filed in the prior application extending the term for response is filed with the papers constituting the filing of the Continuation Application. Notice of November 5, 1985 (1060 O.G. 27).*

☐ A copy of the petition for extension of time in the prior application is attached.

17. Conditional Petition for Extension of Time in Prior Application

(complete this item and file conditional petition in the prior application if previous item not applicable)

☐ A conditional petition for extension of time is being filed in the pending parent application.

NOTE: *The PTO finds it useful if a copy of the petition filed in the prior application extending the term for response is filed with the paper constituting the filing of the continuation application. Notice of Nov. 5, 1985 (1060 O.G. 27).*

☐ A copy of the conditional petition for extension of time in the prior application is attached.

18. Abandonment of Prior Application (*if applicable*)

WARNING: *Do not complete this item if the application being filed is a divisional of the prior application that is not being abandoned.*

NOTE: *"A registered attorney or agent acting under the provisions of § 1.34(a), or of record, may also expressly abandon a prior application as of the filing date granted to a continuing application when filing such a continuing application." 37 CFR 1.138.*

☐ Please abandon the prior application at a time while the prior application is pending or when the petition for extension of time or to revive in that application is granted and when this application is granted a filing date so as to make this application copending with said prior application.

19. Notification in Parent Application of the Filing of This Continuation Application

☒ A notification of the filing of this continuation is being filed in the parent application from which this application claims priority under 35 U.S.C. § 120.

RAB97-002

20. Statement by Assignee (*if applicable*)

☐ In accordance with 37 CFR 3.73, I have reviewed the evidentiary documents establishing my/our ownership of the application identified herein, and certify that to the best of my/our knowledge and belief, title is with me/us who seek to take action.

☐ Assignment submitted herewith for recordal

I hereby declare further that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

J. Nicholas Gross
(type or print name of person signing declaration)

August 4, 1997
Date

J. Nicholas Gross
Signature

993 Highland Circle
P.O. Address of Signatory
Los Altos, CA 94024

☐ Inventor
☐ Assignee of complete interest
☐ Person authorized to sign on behalf of assignee
☒ Attorney or agent of record
☐ Filed under Rule 34(a)

Tel. No. :( )
Reg. No.
(if applicable)

(complete the following, if applicable)

(type name of assignee)

Title of person authorized to sign on behalf of assignee

Address of assignee

Assignment recorded in PTO on

Reel _____

Frame _____

The statement under 37 C.F.R. 3.73(b)

☐ has been filed in the parent application.

☐ a copy of the statement previously filed in the parent application is attached.

RAB97-002

ORIGINAL

Express Mail Label No. EG214043430US

Improved Address Translation Mechanism and Method in a Computer System

FIELD OF THE INVENTION

The invention relates to the field of address translation for memory management in a computer system.

BACKGROUND OF THE INVENTION

Advanced computer hardware systems operate with complex computer software programs. Computer system designers typically separate the virtual address space, the address space used by programmers in their development of software, and the physical address space, the address space used by the computer system. This separation allows programmers to think in terms of their conceptual models, and to design computer software programs without reference to specific hardware implementations. During the actual execution of programs by the computer system, however, these separate addresses must be reconciled by translating software program virtual addresses into actual physical addresses that can be accessed in a computer memory subsystem.

There are many well known approaches for address translation in the memory management mechanism of a computer system. These approaches fall into basically two major categories: those which map the smaller virtual (sometimes called logical, symbolic or user) addresses onto larger physical or real memory addresses, and those which map larger virtual addresses onto smaller physical memory. Translation mechanisms of the former category are employed typically in minicomputers in which relatively small address fields (*e.g.*: 16 bit addresses) are mapped onto larger real memory. Translation mechanisms of the second category are used typically in microprocessors, workstations and mainframes. Within each of these categories, segmentation only, paging only, and a combination of segmentation and paging are well known for accomplishing the translation process.

The present invention is primarily directed to address translation mechanisms where larger virtual addresses are mapped onto smaller physical addresses, and further to systems where segmentation and optional paging is employed.

In a segmentation portion of an address translation system, the address space of a user program (or programs cooperatively operating as processes or tasks), is regarded as a collection of segments which have common high-level properties, such as code, data, stack, etc. The segmented address space is referenced by a 2-tuple, known as a virtual address, consisting of the following fields: <<s>,<d>>, where <s> refers to a segment number (also called identifier or locator), and <d> refers to a displacement or offset, such as a byte displacement or offset, within the segment identified by the segment number. The virtual address <17,421>, for example, refers to the 421st byte in segment 17. The segmentation portion of the address translation mechanism, using information created by the operating system of the computer system, translates the virtual address into a linear address in a linear address space.

In a paging portion of an address translation system, a linear (or intermediate) address space consists of a group of pages. Each page is the same size (*i.e.* it contains the same number of addresses in the linear space). The linear address space is mapped onto a multiple of these pages, commonly, by considering the linear address space as the 2-tuple consisting of the following fields: <<page number >,< page offset>>. The page number (or page frame number) determines which linear page is referenced. The page offset is the offset or displacement, typically a byte offset, within the selected page.

In a paged system, the real (physical) memory of a computer is conceptually divided into a number of page frames, each page frame capable of holding a single page. Individual pages in the real memory are then located by the address translation mechanism by using one or more page tables created for, and maintained by, the operating system. These page tables are a mapping from a page number to a page frame. A specific page may or may not be present in the real memory at any point in time.

Address translation mechanisms which employ both segmentation and paging are well known in the art. There are two common subcategories within this area of virtual address translation schemes: address translation in which paging is an integral part of the segmentation mechanism; and, address translation in which paging is independent from segmentation.

In prior art address translation mechanisms where paging is an integral part of the segmentation mechanism, the page translation can proceed in parallel with the segment translation since segments must start at page boundaries and are fixed at an integer number of pages. The segment number typically identifies a specific page table and the segment offset identifies a page number (through the page table) and an offset within that page. While this mechanism has the advantage of speed (since the steps can proceed in parallel) it is not flexible (each segment must start at a fixed page boundary) and is not optimal from a space perspective (*e.g.* an integer number of pages must be used, even when the segment may only spill over to a fraction of another page).

In prior art address translation mechanisms where paging is independent from segmentation, page translation generally cannot proceed until an intermediate, or linear, address is first calculated by the segmentation mechanism. The resultant linear address is then mapped onto a specific page number and an offset within the page by the paging mechanism. The page number identifies a page frame through a page table, and the offset identifies the offset within that page. In such mechanisms, multiple segments can be allocated into a single page, a single segment can comprise multiple pages, or a combination of the above, since segments are allowed to start on any byte boundary, and have any byte length. Thus, in these systems, while there is flexibility in terms of the segment/page relationship, this flexibility comes at a cost of decreased address translation speed.

Certain prior art mechanisms where segmentation is independent from paging allow for optional paging. The segmentation step is always applied, but the paging step is either performed or not performed as selected by the operating system.

These mechanisms typically allow for backward compatibility with systems in which segmentation was present, but paging was not included.

Typical of the prior art known to the Applicant in which paging is integral to segmentation is the Multics virtual memory, developed by Honeywell and described by the book, "The Multics System", by Elliott Organick. Typical of the prior art known to the Applicant in which optional paging is independent from segmentation is that described in U.S. Pat. No. 5,321,836 assigned to the Intel Corporation, and that described in the Honeywell DPS-8 Assembly Instructions Manual. Furthermore, U.S. Patent 4,084,225 assigned to the Sperry Rand Corporation contains a detailed discussion of general segmentation and paging techniques, and presents a detailed overview of the problems of virtual address translation.

Accordingly, a key limitation of the above prior art methods and implementations where segmentation is independent from paging is that the linear address must be fully calculated by the segmentation mechanism each time before the page translation can take place for each new virtual address. Only subsequent to the linear address calculation, can page translation take place. In high performance computer systems computer systems, this typically takes two full or more machine cycles and is performed on each memory reference. This additional overhead often can reduce the overall performance of the system significantly.

SUMMARY OF THE INVENTION

An object of the present invention, therefore, is to provide the speed performance advantages of integral segmentation and paging and, at the same time, provide the space compaction and compatibility advantages of separate segmentation and paging.

A further object of the present invention is to provide a virtual address translation mechanism which architecturally provides for accelerating references to main memory in a computer system which employs segmentation, or which employs both segmentation and optional paging.

Another object of the present invention is to provide additional caching of page information in a virtual address translation scheme.

An further object of the present invention is to provide a virtual address translation mechanism which reduces the number of references required to ensure memory access.

According to the present invention, a segmentation unit converts a virtual address consisting of a segment identifier and a segment offset into a linear address. The segmentation unit includes a segment descriptor memory, which is selectable by the segment identifier. The entry pointed to by the segment identifier contains linear address information relating to the specific segment (i.e., linear address information describing the base of the segment referred to by the segment identifier, linear address information describing the limit of the segment referred to by the segment identifier, etc.) as well as physical address information pertaining to the segment - such as the page base of at least one of the pages represented by said segment.

In the above embodiment, unlike prior art systems, both segmentation and paging information are kept in the segmentation unit portion of the address translation system. The caching of this page information in the segmentation unit permits the address translation process to occur at much higher speed than in prior art systems, since the physical address information can be generated without having to perform a linear to physical address mapping in a separate paging unit.

The page base information stored in the segmentation unit is derived from the page frame known from the immediately prior in time address translation on a segment-by-segment basis. In order to complete the full physical address translation (*i.e.*, a page frame number and page offset), the segmentation unit combines the page frame from the segment descriptor memory with the page offset field, and may store this result in a segmentation unit memory, which can be a memory table, or a register, or alternatively, it may generate the full physical address on demand.

This fast physical address generated by the segmentation unit based on the virtual address and prior page information can be used by a bus interface to access a

physical location in the computer memory subsystem, even before the paging unit has completed its translation of the linear address into a page frame and page offset. Thus, fewer steps and references are required to create a memory access. Consequently, the address translation step occurs significantly faster. Since address translation occurs in a predominant number of instructions, overall system performance is improved.

The memory access is permitted to proceed to completion unless a comparison of the physical address information generated by the paging unit with the fast physical address generated by the segmentation unit shows that the page frame information of the segmentation unit is incorrect.

In alternative embodiments, the segmentation unit either generates the page offset by itself (by adding the lower portion of the segment offset and the segment base address) or receives it directly from the paging unit.

In further alternate embodiments, the incoming segment offset portion of the virtual address may be presented to the segmentation unit as components. The segmentation unit then combines these components in a typical base-plus-offset step using a conventional multiple input (typically 3-input) adder well known in the prior art.

As shown herein in the described invention, the segment descriptor memory may be a single register, a plurality of registers, a cache, or a combination of cache and register configurations.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a typical prior art virtual address translation mechanism using segmentation and independent paging.

Figure 2A is a detailed diagram of a typical segment descriptor register of the prior art.

Figure 2B is a detailed diagram of an embodiment of the present invention, including a portion of a segment descriptor memory used for storing physical address information;

Figure 3A is a block diagram of an embodiment of the present invention employing segmentation and optional paging, and showing the overall structure and data paths used when paging is disabled during an address translation;

Figure 3B is a block diagram of the embodiment of Figure 3A showing the overall structure and data paths used when paging is enabled during an address translation;

Figure 3C is a block diagram of another embodiment of the present invention, showing an alternative circuit for generating the fast physical address information.

DETAILED DESCRIPTION OF THE INVENTION

General Discussion of Paging & Segmentation

The present invention provides for improved virtual address translation in a computer system. The preferred embodiment employs the invention in a single-chip microprocessor system, however, it is well understood that such a virtual address translation system could be implemented in multiple die and chip configurations without departing from the spirit or claims of the present invention.

Before embarking on a specific discussion of the present invention, however, a brief explanation of the general principles of segmentation and paging follows in order to provide additional background information, and so that the teachings of the present invention may be understood in a proper context.

Referring to FIG.1, a typical prior art virtual address translation mechanism 100 using both segmentation and, optionally, paging in a computer system is shown. As described in this figure, a data path within the microprocessor transmits a virtual address 101, consisting of segment identifier 101a and a segment offset 101b, to segmentation unit 130. Segments are defined by segment descriptor entries in at least one segment descriptor table or segment descriptor segment (not shown). Segment descriptor tables are created and managed by the operating system of the computer system, and are usually located in the memory subsystem. Segment descriptor entries are utilized in the CPU of the computer system by loading them

into segment descriptor register 190 or a segment descriptor cache (not shown); the segment descriptor register/cache is usually internal to the CPU, and thus more quickly accessible by the translation unit.

In the paging unit 150, pages are defined by a page table or multiple page tables (not shown), also created and managed by the operating system; again, these tables are also typically located in a memory subsystem. All or a portion of each page table can be loaded into a page cache 107 (within the CPU, sometimes called a translation look-aside buffer) to accelerate page references.

In operation, the segmentation unit 130 first translates a virtual address to a linear address and then (except in the case when optional paging is disabled) paging unit 150 translates the linear address into a real (or physical) memory address.

Typically (as in an x86 microprocessor) the segmentation unit translates a 48-bit virtual address 101 consisting of a 16-bit segment identifier (<s>) 101a and a 32-bit displacement within that segment (<d>) 101b to a 32-bit linear (intermediate) address 106. The 16-bit segment identifier 101a uniquely identifies a specific segment; this identifier is used to access an entry in a segment descriptor table (not shown). In the prior art, this segment descriptor entry contains a base address of the segment 191, the limit of the segment 192, and other attribute information described further below. The segment descriptor entry is usually loaded into a segment descriptor register 190.

Using adder 105, the segmentation unit adds the segment base 191 of the segment to the 32-bit segment offset 101b in the virtual address to obtain a 32-bit linear address. The 32-bit segment offset 101b in the virtual address is also compared against the segment limit 192, and the type of the access is checked against the segment attributes. A fault is generated and the addressing process is aborted if the 32-bit segment offset is outside the segment limit, or if the type of the access is not allowed by the segment attributes.

The resulting linear address 106 can be treated as an offset within a linear address space; and in the commonly implemented schemes of the prior art, these offsets are frequently byte offsets. When optional paging is disabled, the linear

address 106 is exactly the real or physical memory address 108. When optional paging is enabled, the linear address is treated as a 2- or 3-tuple depending on whether the paging unit 150 utilizes one or two level page tables.

In the 2-tuple case shown in FIG.1, which represents single level paging, the linear address, <<p>,<pd>> is divided into a page number field <p> 106a, and a page displacement (page offset) field within that page (<pd>) 106b. In the 3-tuple case (not shown) <<dp>,<p>,<pd>>, the linear address is divided into a page directory field (<dp>), a page number field <p> and a page displacement field <pd>. The page directory field indexes a page directory to locate a page table (not shown). The page number field indexes a page table to locate the page frame in real memory corresponding to the page number, and the page displacement field locates a byte within the selected page frame. Thus, paging unit 150 translates the 32-bit linear address 106 from the segmentation unit 130 to a 32-bit real (physical) address 108 using one or two level page tables using techniques which are well known in the art.

In all of the above prior art embodiments where segmentation is independent from paging, the segment descriptor table or tables of the virtual address translator are physically and logically separate from the page tables used to perform the described page translation. There is no paging information in the segment descriptor tables and, conversely, there is no segmentation information in the page tables.

This can be seen in FIG. 2A. In this figure, a typical prior art segment descriptor entry 200, is shown as it is typically used in a segment descriptor table or segment descriptor register associated with a segmentation unit. As can be seen there, segment descriptor 200 includes information on the segment base 201, the segment limit 202, whether the segment is present (P) 203 in memory, the descriptor privilege level (DPL) 204, whether the segment belongs to a user or to the system (S) 205, and the segment type 206 (code, data, stack, etc.)

For additional discussions pertaining to the prior art in segmentation, paging, segment descriptor tables, and page tables, the reader is directed to the references U.S. Patent Nos. 5,408,626, 5,321,836, 4,084,225, which are expressly incorporated by reference herein.

Improved Segmentation Unit Using Paging Information

As shown in the immediate prior art, the paging and segmentation units (circuits) are completely separate and independent. Since the two units perform their translation sequentially, that is, the segment translation must precede the page translation to generate the linear address, high performance computer systems, such as those employing superscalar and superpipelined techniques, can suffer performance penalties. In some cases, it is even likely that the virtual address translation could fall into the systems' "critical path". The "critical path" is a well-known characteristic of a computer system and is typically considered as the longest (in terms of gate delay) path required to complete a primitive operation of the system.

Accordingly, if the virtual address translation is in the critical path, the delays associated with this translation could be significant in overall system performance. With the recognition of this consideration, the present invention includes page information in the segment translation process. The present invention recognizes the potential performance penalty of the prior art and alleviates it by storing paging information in the segmentation unit obtained from a paging unit in previous linear-to-real address translations.

As can be seen in FIG. 2B, the present invention extends the segment descriptor entries of the prior art with a segment entry 290 having two additional fields: a LAST PAGE FRAME field 297 and a VALID field 298. The LAST PAGE FRAME field 297 is used to hold the high-order 20 bits (*i.e.*: the page frame) of the real (physical) memory address of the last physical address generated using the specified segment identifier. The VALID field 298 is a 1-bit field, and indicates whether or not the LAST PAGE FRAME field 297 is valid. The remaining fields

291-296 perform the same function as comparable fields 201-206 respectively described above in connection with FIG. 2A.

Segment descriptor tables (not shown) can be located in a memory subsystem, using any of the techniques well-known in the art. As is also known in the art, it is possible to speed up address translation within the segmentation unit by using a small cache, such as one or more registers, or associative memory. The present invention makes use of such a cache to store segment entries 290 shown above. Unlike the prior art, however, the segment entries 290 of the present invention each contain information describing recent physical address information for the specified segment. Accordingly, this information can be used by a circuit portion of the segmentation unit to generate a new physical address without going through the linear to physical mapping process typically associated with a paging unit.

While in some instances the physical address information may change between two time-sequential virtual addresses to the same segment (and thus, a complete translation is required by both the segmentation and paging units), in the majority of cases the page frame information will remain the same. Thus, the present invention affords a significant speed advantage over the prior art, because in the majority of cases a complete virtual-linear-physical address translation is not required before a memory access is generated.

Embodiment With Segmentation & Optional Paging/Paging Disabled

Referring to FIG. 3A, the advantage of using this new information in segment entry 290 in a segmentation unit or segmentation circuit is apparent from a review of the operation of an address translation. In this figure, a paging unit (or paging circuit) is disabled, as for example might occur only when a processor is used in a real mode of operation, rather than a protected mode of operation.

In a preferred embodiment, the present invention employs a segment descriptor memory comprising at least one, and preferably many, segment descriptor registers 390, which are identical in every respect to the segment descriptor register described above in connection with Fig.2B. These segment descriptor registers are

loaded from conventional segment descriptor tables or segment descriptor segments which are well known in the art. Each segment descriptor register 390 is loaded by the CPU before it can be used to reference physical memory. Segment descriptor register 390 can be loaded by the operating system or can be loaded by application programs. Certain instructions of the CPU are dedicated to loading segment descriptor registers, for example, the "LDS" instruction, "Load Pointer to DS Register". Loading by the operating system, or execution of instructions of this type, causes a base 391, limit 392, descriptor privilege level 394, system/user indicator 395, and type 396 to be loaded from segment tables or segment descriptor segments as in the prior art. The three remaining fields are present 393, LAST PAGE FRAME 397 and VALID 398. When a segment descriptor register 390 is loaded, present 393 is set to 1, indicating that the segment descriptor register 390 contents are present; the valid field 398 is set to 0, indicating that the last page frame number field 397 is not valid; and the LAST PAGE FRAME field 397 is not set, or may be set to 0.

After the loading of a segment descriptor register 390, instructions of the CPU may make references to virtual memory; if a segment descriptor register is referenced before it is loaded, as indicated by present field 393 set to 0, a fault occurs and the reference to the segment descriptor register is aborted.

As explained above, the CPU makes references to virtual memory by specifying a 48-bit virtual address, consisting of a 16-bit segment identifier 301a and a 32-bit segment offset 301b. A data path within the CPU transmits virtual address 301 to the address translation mechanism 300.

Segment descriptor memory 390 is indexed by segment number, so each entry in this memory containing data characteristics (*i.e.*, base, access rights, limit) of a specific segment is selectable by the segment identifier from the virtual address. Assuming this is the first reference to physical memory specifying a newly loaded segment descriptor register, since the VALID bit 398 is set to false, a prior art virtual address translation takes place. This involves, among other things, as explained earlier, various validity checks (including checking attributes 394-396,

segment limit checking using comparator 302 and potentially others), and using adder 305 to add the segment descriptor's base address 391 to the segment offset 301b to calculate a linear address 306.

While the implementation in the embodiment of Fig. 3A shows the addition of the base address 391 to the segment offset 301b using adder 305 to generate the linear address 306, it will be understood by those skilled in the art that this specific implementation of the virtual to linear address translation is not the only implementation of the present invention. In other implementations, the segment offset 301b might consist of one or more separate components. Different combinations of one or more of these components might be combined using well known techniques to form a linear address, such as one utilizing a three-input adder. The use of these components is discussed, for example, in U.S. Patent No. 5,408,626, and that description is incorporated by reference herein.

As is well known, in this embodiment where paging is disabled, linear address 306 is also a physical address which can be used as the physical address 308. Memory access control operations are not shown explicitly since they are only ancillary to the present invention, and are well described in the prior art. In general, however, a bus interface unit 380 is typically responsible for interactions with the real (physical) memory subsystem. The memory subsystem of a computer system employing the present invention preferably has timing and address and data bus transaction details which are desirably isolated from the CPU and address translation mechanism. The bus interface unit 380 is responsible for this isolation, and can be one of many conventional bus interface units of the prior art.

In the present invention, bus interface unit 380 receives the real memory address 308 from address translation mechanism 300 and coordinates with the real memory subsystem to provide data, in the case of a memory read request, or to store data, in the case of a memory write request. The real memory subsystem may comprise a hierarchy of real memory devices, such as a combination of data caches and dynamic RAM, and may have timing dependencies and characteristics which are

isolated from the CPU and address translation mechanism 300 of the computer system by the bus interface unit 380.

Simultaneous with the first memory reference using the calculated physical address 308, the LAST PAGE FRAME field of the selected segment descriptor register 390 is loaded with the high-order 20 bits of the physical address, i.e.: the physical page frame, and the VALID bit is set to indicate a valid state. This paging information will now be used in a next virtual address translation.

Accordingly, when a next, new virtual address 301 is to be translated, the entry selected from segment descriptor memory 390 will likely contain the correct physical frame page number (in the LAST PAGE FRAME field 397). Thus, in most cases, the base physical address in memory for the next, new referenced virtual address will also be known from a previous translation.

The first step of the virtual address translation, therefore, is to determine if a FAST PHYSICAL ADDRESS 303 can be used to begin a fast physical memory reference. Adder 309, a 12-bit adder, adds the low-order 12 bits of the segment offset 301b of virtual address 301 to the low-order 12-bits of base 391 of the segment entry in segment descriptor register 390 referenced by the segment identifier 301a. This addition results in a page offset 303b. In parallel with adder 309, 32-bit adder 305 begins a full 32-bit add of segment base 391 and segment offset 301b, to begin producing the linear address; however, this full 32-bit add will obviously require more time. In the preferred embodiment, adder 309 is a separate 12-bit adder; however, it should be noted that adder 309 also could be implemented as the low order 12-bits of 32-bit adder 305.

Simultaneous with the beginning of these two operations, VALID bit 398 is inspected. If VALID bit 398 is set to 1, as soon as 12-bit adder 309 has completed, 20-bit LAST PAGE FRAME 397 is concatenated with the result of adder 309 to produce FAST PHYSICAL ADDRESS 303, consisting of a page frame number 303a, and page offset 303b. FAST PHYSICAL ADDRESS 303 then can be used to tentatively begin a reference to the physical memory. It should be understood that

the FAST PHYSICAL ADDRESS 303 transmitted to bus interface unit 380 could also be stored in a register or other suitable memory storage within the CPU.

In parallel with the fast memory reference, limit field 392 is compared to the segment offset 301b of the virtual address by comparator 302. If the offset in the virtual address is greater than the limit, a limit fault is generated, and virtual address translation is aborted.

Also in parallel with the fast memory reference, adder 305 completes the addition of base 391 to the segment offset field 301b of virtual address to produce linear address (in this case physical address also) 306. When this calculation is completed, the page frame number 308a of physical address 308 is compared to LAST PAGE FRAME 397 by Not Equal Comparator 304. If page frame 308a is unequal to the LAST PAGE FRAME 397, or if 12-bit Adder 309 overflowed (as indicated by a logic "1" at OR gate 310), the fast memory reference is canceled, and the linear address 306, which is equal to the physical address 308, is used to begin a normal memory reference. If page frame 308a is equal to LAST PAGE FRAME 397, and 12-bit Adder 309 did not overflow (the combination indicated by a logic "0" at the output of OR gate 310), the fast memory reference is allowed to fully proceed to completion.

After any fast memory reference which is cancelled by the CANCEL FAST PHYSICAL ADDRESS signal output of OR gate 310, page frame 308a is loaded into the LAST PAGE FRAME 397 in the segment descriptor memory 390 for subsequent memory references.

Depending on the particular design desired, it should also be noted that writes to the memory, or reads which cause faults using FAST PHYSICAL ADDRESS 303 may be pended since the FAST PHYSICAL ADDRESS 303 may prove to be invalid.

Accordingly, it can be seen that the parallel physical address calculation undertaken by the improved segmentation unit of the present invention generates a faster physical memory access than possible with prior art systems.

Embodiment With Segmentation & Paging/Paging Enabled

The present invention can also be used with address translation units using paging enabled, as can be seen in the embodiments of FIGs. 3B and 3C.

In the embodiment of FIG. 3B, the same segmentation unit structure 300 as that shown in FIG. 3A is used, and the operation of segmentation unit 300 is identical to that already explained above. As before, segment descriptor memory (registers) 390 are loaded from conventional segment descriptor tables or segment descriptor segments, using one or more of the procedures described above. First, the base 391 limit 392 descriptor privilege level 394, system/user indicator 395, and type 396 are loaded from segment tables or segment descriptor segments as explained earlier. When segment descriptor register 390 is loaded, present 393 is set to 1, indicating that the segment descriptor register 390 contents are present; the valid field 398 is set to 0, indicating that the last page frame number field 397 is not valid; and the LAST PAGE FRAME field 397 is not set, or may be set to 0.

As explained above, after the loading of a segment descriptor register 390, instructions of the CPU may make references to virtual memory; if a segment descriptor register is referenced before it is loaded, as indicated by present field 393 set to 0, a fault occurs and the reference to the segment descriptor register is aborted.

As further explained above, the 48 bit virtual address 301 (consisting of a 16 bit segment identifier 301a and a 32 bit segment offset 301b) is transmitted by a data path to segmentation unit 300, and an index into segment descriptor memory 390 is performed to locate the specific segment descriptor for the segment pointed to by segment identifier 301a. Assuming this is the first reference to physical memory specifying a newly loaded segment descriptor register, since the VALID bit is set to false, a prior art virtual address translation takes place. This involves, among other things, as explained earlier, various validity checks (including checking attributes 394-396, segment limit checking using comparator 302 and potentially others), and using adder 305 to add the segment descriptor's base address 391 to the segment offset 301b to calculate a linear address 306.

As is well known, in this configuration where paging is enabled, linear address 306 must undergo a further translation by paging unit 350 to obtain the physical address 308 in the memory subsystem. In the preferred embodiment of the invention, looking first at FIG.3B, the output of adder 305 will be a 32-bit linear address, corresponding to a 20-bit page number 306a and a 12-bit page offset 306b. Typically, the page number 306a is then indexed into a page descriptor table (not shown) to locate the appropriate page frame base physical address in memory. These page descriptor tables are set up by the operating system of the CPU using methods and structures well known in the art, and they contain, among other things, the base physical address of each page frame, access attributes, etc.

However, in most systems, including the present invention, a page cache 307 is used in order to hold the physical base addresses of the most recently used page frames. This cache can take the form of a table, associative cache, or other suitable high speed structure well known in the art. Thus, page number 306a is used to access page data (including physical base addresses for page frames) in an entry in page cache 307.

If page cache 307 hits, two things happen: first, a 20-bit PAGE FRAME 307a (the page frame in physical memory) replaces the high-order 20 bits (page number 306a) of the linear address 306, and, when concatenated with the page offset 306b results in a real (physical) address 308, which is used to perform a memory access through bus interface unit 380 along the lines explained above. Second, newly generated page frame 308a is also stored in segment descriptor memory 390 in the selected LAST PAGE FRAME field 397 to be used for a fast access in the next address translation. When LAST PAGE FRAME field 397 is stored, selected VALID bit 398 is set to 1 to indicate that LAST PAGE FRAME 397 is valid for use.

In the event of a page cache miss, the appropriate page frame number 308a is located (using standard, well-known techniques) to generate physical address 308, and is also loaded into segment descriptor memory 390 in the selected LAST PAGE FRAME field 397. The selected VALID bit 398 is also set to indicate a valid state.

Thus, there is paging information in the segmentation unit that will now be used in the next virtual address translation.

When a next, new virtual address 301 is to be translated, the segment identifier 301a will likely be the same as that of a previously translated virtual address, and the entry selected from segment descriptor memory 390 will also likely contain the correct physical frame (in LAST PAGE FRAME field 397) from the previous translation. As with the above embodiment, one or more registers, or a cache may be used for the segment descriptor memory 390.

The first step then determines if a FAST PHYSICAL ADDRESS 303 can be used to begin a fast physical memory reference. Adder 309, a 12-bit adder, adds the low-order 12 bits of the segment offset 301b of virtual address 301 to the low-order 12-bits of base 391 of the segment entry in segment descriptor register 390 referenced by the segment identifier 301a. This addition results in a page offset 303b. In parallel with adder 309, 32-bit adder 305 begins a full 32-bit add of segment base 301 and segment offset 301b, to begin producing the linear address; however, this full 32-bit add will obviously require more time. In the preferred embodiment, adder 309 is a separate 12-bit adder; however, it should be noted that adder 309 also could be implemented as the low order 12-bits of 32-bit adder 305.

Simultaneous with these beginning of these two operations, VALID bit 398 is inspected. If VALID bit 398 is set to 1, as soon as 12-bit adder 309 has completed, 20-bit LAST PAGE FRAME 397 is concatenated with the result of adder 309 to produce FAST PHYSICAL ADDRESS 303, consisting of a page frame number 303a, and page offset 303b. FAST PHYSICAL ADDRESS 303 then can be used to tentatively begin a reference to the physical memory. Again, it should be understood that the FAST PHYSICAL ADDRESS 303 transmitted to bus interface unit 380 could also be stored in a register or other suitable memory storage within the CPU.

As before, limit field 302 is compared to the segment offset 301b of the virtual address by comparator 302. If the offset in the virtual address is greater than the limit, a limit fault is generated, and virtual address translation is aborted.

This new virtual address is also translated by paging unit 350 in the same manner as was done for the previous virtual address. If page cache 307 hits based on the page number 306a, two things happen: first, a 20-bit PAGE FRAME 307a (the page frame in physical memory) replaces the high-order 20 bits (page number 306a) of the linear address 306, and, when concatenated with the page offset 306b results in a physical address 308. This real address may or may not be used, depending on the result of the following: in parallel with the aforementioned concatenation, the PAGE FRAME 307a, is compared to LAST PAGE FRAME 397 from the segment descriptor memory 390 by Not Equal Comparator 304. The result of Not Equal Comparator (that is, the Not Equal condition) is logically ORed with the overflow of 12-bit adder 309 by OR gate 310. If the output of OR gate 310 is true (i.e. CANCEL FAST PHYSICAL ADDRESS is equal to binary one), or if PAGE CACHE 307 indicates a miss condition, the fast memory reference previously begun is canceled, since the real memory reference started is an invalid reference. Otherwise, the fast memory reference started is allowed to fully proceed to completion, since it is a valid real memory reference.

If CANCEL FAST PHYSICAL ADDRESS is logical true, it can be true for one of two, or both reasons. In the case that Or gate 310 is true, but page cache 307 indicates a hit condition, physical address 308 is instead used to start a normal memory reference. This situation is indicative of a situation where LAST PAGE FRAME 397 is different from the page frame 308a of the current reference.

In the case that page cache 307 did not indicate a hit, a page table reference through the page descriptor table is required and virtual address translation proceeds as in the prior art. The page frame 308a information is again stored in the LAST PAGE FRAME field 397 in the segment descriptor memory 390 for the next translation.

Also, after any fast memory reference which is canceled by the CANCEL FAST PHYSICAL ADDRESS signal output of OR gate 310, page frame number 308a is loaded into the LAST PAGE FRAME 397 in the segment descriptor memory 390 for subsequent memory references.

Depending on the particular design desired, it should also be noted that in this embodiment also, writes to the memory, or reads which cause faults using FAST PHYSICAL ADDRESS 303 may be pended since the FAST PHYSICAL ADDRESS 303 may prove to be invalid.

The alternative embodiment shown in FIG. 3C is identical in structure and operation to the embodiment of FIG. 3B, with the exception that the 12-bit Adder 309 is not employed. In this embodiment, the segmentation unit 330 does not create the lower portion (page offset 303a) of the fast physical address in this manner. Instead, the page offset 306a resulting from 32-bit adder 305 is used.

It can be seen that the present invention has particular relevance to computers using sequential type of segmentation and paging translation, such as the X86 family of processors produced by the Intel Corporation (including the Intel 80386, Intel 80486 and the Intel Pentium Processor), other X86 processors manufactured by the NexGen Corporation, Advanced Micro Devices, Texas Instruments, International Business Machines, Cyrix Corporation, and certain prior art computers made by Honeywell. These processors are provided by way of example, only, and it will be understood by those skilled in the art that the present invention has special applicability to any computer system where software executing on the processors is characterized by dynamic execution of instructions in programs in such a way that the virtual addresses are generally logically and physically located near previous virtual addresses.

The present invention recognizes this characteristic, employing acceleration techniques for translating virtual to real addresses. In particular, the present invention utilizes any of the commonly known storage structures (specific examples include high speed registers and/or caches) to store previous address translation information, and to make this previous address translation information available to the system whenever the next subsequent reference relies on the same information. In this way, the system can utilize the previously stored information from the high speed storage to begin real memory references, rather than be forced to execute a

more time consuming translation of this same information, as was typically done in the prior art.

As will be apparent to those skilled in the art, there are other specific circuits and structures beyond and/or in addition to those explicitly described herein which will serve to implement the translation mechanism of the present invention. Finally, although the above description enables the specific embodiment described herein, these specifics are not intended to restrict the invention, which should only be limited as defined by the following claims.

I claim:

1. A circuit for storing address translation information, said circuit comprising:
   a) a data path for receiving a virtual address, said virtual address including a segment identifier and a segment offset; and
   b) a segment descriptor memory coupled to said data path and selectable by said segment identifier, said memory capable of storing at least the following:
      I) linear address information describing the base of the segment,
      ii) linear address information describing the limit of the segment, and
      iii) a page frame describing at least a portion of a physical address of said segment.

2. The circuit of claim 1, wherein the segment descriptor memory is one or more registers.

3. The circuit of claim 1, wherein the segment descriptor memory is a cache.

4. The circuit of claim 1, further including a physical address register coupled to the segment descriptor memory for storing a physical address, said physical address

being comprised of the page frame from said segment descriptor memory and a page offset.

5. The circuit of claim 4, wherein the physical address stored in the physical address register is used to perform a memory access.

6. The circuit of claim 1, wherein the segment descriptor memory is further capable of storing:

(iv) information describing whether said page frame can be used for an address translation.

7. In a system having both a segmentation unit and a paging unit for translating physical addresses from virtual addresses, said virtual addresses including a segment identifier and segment offset, the improvement wherein:

the segmentation unit includes a segment descriptor memory selectable by said segment identifier, said memory capable of storing at least the following:

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment, and

iii) physical address information relating to said segment.

8. A circuit for storing address translation information in a computer system, said computer using both segmentation and paging to translate a virtual address having a segment identifier and segment offset into a physical address, said circuit comprising:

a segment descriptor memory selectable by said segment identifier, said memory capable of storing at least the following:

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment, and

iii) page frame describing at least a portion of a physical address of said segment.

9. An address translation system for translating a virtual address having a segment identifier and an offset field into a physical address, said address translation system being used in a computer system and comprising:

(a) a segmentation unit for generating linear address information based on the virtual address information, and for storing first physical address information having a first physical page frame and a first physical page offset;

(b) a paging unit coupled to the segmentation unit for generating said first physical page frame and second physical address information having a second physical page frame and a second page offset; and

© said segmentation unit further including:

[1] a segment descriptor memory, selectable by said segment identifier of said virtual address, and capable of storing

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment,

iii) said first physical page frame;

[2] a limit comparator for comparing whether the offset field exceeds the limit of the segment;

wherein a system memory access can be made by said computer system based on said first physical address information.

10. The system of claim 9, wherein the segmentation unit further includes a physical address comparator for comparing the first physical page frame and the second physical page frame, and wherein the memory access is canceled only if the first physical page frame and second physical page frame are not equal.

11. The system of claim 9, wherein the segmentation unit further includes an adder, and the first physical page offset can be generated by adding a portion of the virtual address offset field and a portion of the segment base in the segment descriptor memory.

12. The system of claim 9, wherein the first physical page offset can be generated by adding the virtual address offset field and the segment base in the segment descriptor memory.

13. The system of claim 9, wherein the paging unit further includes a page cache for storing page frames of physical pages most recently used by the computer system.

14. The system of claim 9, wherein the segment descriptor memory comprises one at least a register and/or cache.

15. The system of claim 9, wherein the first physical page frame generated by the paging unit is stored in the segment descriptor memory of the segmentation unit and can be used in a subsequent virtual to physical address translation.

16. The system of claim 9, wherein the segment descriptor memory is further capable of storing:

(iv) information describing whether said page frame can be used for an address translation.

17. The system of claim 9, wherein the segmentation unit also uses either or both index and displacement information to generate the linear address information.

18. A system for address translation in a CPU comprising:

a) an instruction set employing virtual addresses for accessing data or instructions located at physical addresses in a memory subsystem, said virtual addresses containing a segment identifier field selecting a segment and a segment offset field selecting an offset within the selected segment;

b) a segmentation circuit for generating linear addresses based on the virtual addresses, and for storing physical address information, said segmentation circuit including at least one segment descriptor memory selectable by said segment identifier, said segment descriptor memory capable of storing:

(I) segment data describing a segment, said segment data including linear address information describing the base and limit of the segment in a linear address space;

(ii) a physical address information corresponding to a page frame;

c) a paging circuit for receiving the linear addresses and generating physical addresses, said paging circuit including a page cache, said page cache optionally storing page frame and page offset field information for said CPU; and

d) a bus interface, capable of coupling the segmentation and paging circuits of the CPU to the memory subsystem, and for performing memory accesses in response to physical address information from either of said segmentation and paging circuits.

19. The circuit of claim 18, wherein the segment descriptor memory is one or more registers, or a cache.

20. The system of claim 18, wherein said bus interface provides either separate address/data lines to said memory, or multiplexed address/data lines.

21. The system of claim 18, wherein the segmentation circuit further includes a physical address comparator for comparing the page frame in the segmentation circuit descriptor memory with the page frame in the page cache, and wherein the

memory access is canceled only if the page frame in the segmentation circuit segment descriptor memory is different from the page frame in the page cache.

22. The system of claim 18, wherein the segmentation circuit further includes an adder, and a page offset is generated by adding a portion of the segment offset field to a portion of the segment base in the segment descriptor memory.

23. The system of claim 18, wherein the page frame generated by the paging unit is stored in the segmentation circuit.

24. The system of claim 18 wherein the segment descriptor cache is further capable of storing:

(iv) information describing whether said physical address information can be used for an address translation.

25. The system of claim 18, wherein the segmentation circuit uses index and/or displacement information to generate the linear addresses.

26. An address translation system for translating a virtual address having a segment and an offset field into a physical address, said address translation system being used in a computer system and comprising:

a) a segmentation unit for generating linear address information based on the virtual address information, and for generating and storing a physical address including a page frame and a page offset, said segmentation unit further including:

b) [1] a segment descriptor memory, selectable by said segment identifier of said virtual address, and capable of storing

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment,

iii) said page frame;

[2] a physical address memory for storing said physical address including said page frame number and a page offset; and

wherein an access to a memory subsystem can be made by said computer system based on said physical address.

27. The system of claim 26, wherein the segmentation unit further includes a physical address comparator for comparing the page frame from said virtual address with a page frame corresponding to a different, later in time virtual address, and wherein the memory access is canceled only if the page frames of said different virtual addresses are not equal.

28. The system of claim 26, wherein the segmentation unit further includes an adder, and the page offset is generated by adding a portion of the virtual address offset field and a portion of the segment base in the segment descriptor memory, and then stored in a register corresponding to the first physical address memory.

29. The system of claim 26, wherein the segment descriptor memory includes at least a register and/or a cache.

30. The system of claim 26, further including a bus interface capable of coupling the segmentation unit to the memory subsystem, and for performing memory accesses in response to physical address information from said segmentation unit.

31. The system of claim 30, wherein said bus interface provides either separate address/data lines to said memory, or multiplexed address/data lines.

32. The system of claim 26, wherein the segment descriptor memory is further capable of storing:

(iv) information describing whether said page frame can be used for an address translation.

33. The system of claim 26, wherein the segmentation unit also uses either or both index and displacement information to generate the linear address information.

34. A method of calculating physical addresses from virtual addresses, said method comprising:

a) calculating a first physical address, having a first page frame field and a first page offset field, based on a virtual address;

b) storing said first page frame field of said first physical address;

c) calculating a second physical address based on a second virtual address including a second page frame field and a second page offset field;

d) generating a third physical address based on the first page frame field and the second page offset field;

e) generating a memory access request based on said third physical address.

35. The method of claim 34, further including a step: (f) canceling said access request to memory using said third physical address if the first page frame field is not equal to the second page frame field of said second physical address.

36. The method of claim 34, wherein the page frame fields most recently used by the computer system are stored.

37. The method of claim 34, further including a step: checking whether the first page frame field can be used for an address translation.

ABSTRACT

An improved address translation method and mechanism for memory management in a computer system is disclosed. A segmentation mechanism employing segment registers maps virtual addresses into a linear address space. A paging mechanism optionally maps linear addresses into physical or real addresses. Independent protection of address spaces is provided at each level. Information about the state of real memory pages is kept in segment registers or a segment register cache potentially enabling real memory access to occur simultaneously with address calculation, thereby increasing performance of the computer system.

RAB 95-001

DECLARATION FOR PATENT APPLICATION /POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original. first and sole inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled *Improved Address Translation Mechanism and Method in a Computer System*, the specification of which is attached hereto.

I hereby state that I have reviewed and understand the contents of the above identified specification, including claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56.

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Peter J. Courture, Esq.

J. Nicholas Gross, Esq.

Please address all telephone calls to *Peter J. Courture, Esq.* at telephone number *(415) 968-8885*

address all correspondence to *Peter Courture, Esq., Law+ 993 Highland Circle, Los Altos, CA 94024*

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor (given name, family name) *Richard A. Belgard*

Inventor's signature _____ Date 6/2/95

Residence *21250 Glenmont Drive Saratoga, Ca. 95070*

Citizenship *United States*

Post Office Address *21250 Glenmont Drive Saratoga, Ca. 95070*

RAB 97-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: *Belgard, R.*	Art Unit: *unknown*
Serial No.: *unknown*	Examiner: *unknown*
Filed: *8/4/97 as continuation of parent application serial no. 08/458,479 filed 6/2/95*	
For: *Address Translation Method and Mechanism Using Physical Address Information Including During A Segmentation Process*	

PRELIMINARY AMENDMENT
FOR ACCOMPANYING RULE 1.60 CONTINUATION APPLICATION

Honorable Commissioner
of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

As part of the present filing, the Applicant respectfully submits the following to place this case in condition for allowance:

IN THE TITLE:

Please change the title to: FAST ADDRESS GENERATOR FOR REDUCING VIRTUAL-LINEAR-PHYSICAL ADDRESS CONVERSION TIME.

IN THE SPECIFICATION:

Please insert the following sentences before line 1: "This is a continuation of application serial no. 08/458,479 filed on June 2, 1995. The present application is also related to a further application filed concurrently herewith entitled COMPUTER ADDRESS TRANSLATION USING FAST ADDRESS GENERATOR DURING A SEGMENTATION OPERATION

PERFORMED ON A VIRTUAL ADDRESS, attorney docket no. RAB 97-001."

IN THE CLAIMS:

Please cancel claims 6 - 37.

Please amend claims 1-5 as follows:

1. (Amended) A circuit for storing physical address translation information to reduce address translation time in a computer system, said circuit comprising:

a) a data path for receiving a virtual address, said virtual address including a segment identifier for identifying a segment [identifier] and a segment offset; and

b) a segment descriptor memory coupled to said data path and selectable by said segment identifier, said memory capable of storing at least the following:

i) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment, and

iii) a page frame describing at least a portion of a physical address of said segment; and

wherein a tentative memory reference can be initiated based on the virtual address and the information in the segment descriptor memory and without performing a virtual to linear to physical address translation.

2. (Amended) The circuit of claim 1, wherein the information in the segment descriptor memory is stored in one or more registers, and such information can be combined with a portion of the segment offset to create a physical address that is used to initiate the tentative memory reference.

3. (Amended) The circuit of claim 1, wherein the information in the segment descriptor memory is stored in a cache and such information can be combined with a portion of the segment offset to create a physical address that is used to initiate the tentative memory reference.

4. (Amended) The circuit of claim 1, further including a physical address register coupled to the segment descriptor memory for storing a physical address used to initiate the tentative memory reference, said physical address being comprised of the page frame from said segment descriptor memory and a page offset.

5. (Amended) The circuit of claim [4] 1, wherein the [physical address stored in the physical address register is used to perform a memory access] page frame stored in the segment descriptor memory is based on a prior virtual address.

Please add new claims 38 - 81:

38. A system for performing address translations, said system generating an actual physical address from a virtual address in a time period T, by calculating a linear address based on said virtual address, and by calculating said actual physical address based on said calculated linear address, said system further including:

a fast physical address generator for generating a fast physical address related to said virtual address in a time < T.

39. The system of claim 38, wherein the fast physical address can be used for generating a memory access faster than a memory access based on said actual physical address.

40. The system of claim 39, including a cancellation circuit for cancelling the memory access if the fast physical address and actual physical address are different.

41. The system of claim 38, wherein the fast physical address is generated based on a combination of physical address information from a different virtual address, and partial linear address information relating to said virtual address.

42. The circuit of claim 38, wherein the fast physical address is generated before said calculated linear address.

43. A system for performing address translations using a first operation to convert a first virtual address to a first linear address, and a second operation to convert said first linear address to a first physical address, said system further including:

a tentative physical address generator for generating a tentative physical address related to said first virtual address;

wherein the tentative physical address can be generated before said second operation has completed converting said first linear address.

44. The system of claim 43, wherein the tentative physical address can be used for generating a memory access which is faster than a memory access resulting from said first physical address.

45. The system of claim 44, including a cancellation circuit for cancelling the memory access if the tentative physical address and first physical address are different.

46. The system of claim 43, wherein the tentative physical address is generated based on a combination of prior physical address information and partial linear address information relating to said first virtual address.

47. The circuit of claim 43, wherein the tentative physical address is generated before said first operation has completed converting said first virtual address into said first linear address.

48. The circuit of claim 43, wherein said first virtual address is partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address.

49. A system for performing address translations using a first operation to convert virtual addresses to linear addresses, and a second operation to convert said linear addresses to physical addresses, said system further including:

a fast physical address generator for generating fast physical addresses related to said virtual addresses;

wherein the fast physical addresses can be generated while said virtual addresses are being converted in said first operation into said linear addresses.

50. The system of claim 49, wherein the fast physical addresses can be used for generating memory accesses faster than memory accesses resulting from said calculated physical addresses.

51. The system of claim 50, including a cancellation circuit for cancelling the memory accesses if the fast physical addresses and calculated physical addresses are different.

52. The system of claim 49, wherein the fast physical addresses are generated based on a combination of physical address information and partial linear address information relating to said virtual addresses.

53. The system of claim 49, wherein said virtual addresses are partially converted to linear addresses by the fast physical address circuit and are combined with physical address information relating to prior virtual addresses to generate the tentative physical addresses.

54. A system for performing address translations comprising:

a virtual to linear address converter circuit for generating a calculated linear address based on a virtual address; and

a linear to physical address converter circuit for generating a calculated physical address based on the calculated linear address, the calculated physical address including a calculated page frame and a calculated page offset; and

a fast physical address circuit for generating a fast physical address including a fast page frame and a fast page offset;

wherein a memory reference can be generated based on the fast physical address.

55. The system of claim 54, wherein the fast physical address is based on linear address information relating to the virtual address and physical address information relating to a prior virtual address.

56. The system of claim 54, wherein the virtual address is partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address.

57. A system for performing address translations using a first operation to convert a first virtual address to a first linear address, and a second operation to convert said first linear address to a first physical address, the system further including:

an address translation memory, accessible by said system while said first operation is converting said first virtual address, and capable of storing prior physical address information generated during a prior address translation by said second operation based on a prior virtual address;

wherein a fast physical address can be generated based on the prior physical address information and said first linear address before said second operation has completed converting said first linear address.

58. The system of claim 57, wherein the fast physical address can be used for an accelerated memory access which is faster than a memory access resulting from said first physical address.

59. The system of claim 58, including a cancellation circuit for cancelling the fast memory access if the fast physical address and first physical address are different.

60. The system of claim 57, wherein the fast physical address is comprised of:

(iii) a page frame portion based on the prior physical address information; and

(iv) a page offset portion based on the result of converting said first virtual address to a first linear address.

61. A system for performing address translations comprising:

an address translation memory capable of storing:

(i) a portion of a physical address corresponding to a stored page frame; and

(ii) segment base information relating to a virtual address; and

a virtual to linear address converter circuit for generating a calculated linear address based on combining a portion of the virtual address and the segment base; and

a linear to physical address converter circuit for receiving and generating a calculated physical address based on the calculated linear address, the calculated physical address including a first page frame and a first page offset; and

a fast physical address circuit for generating a fast physical address comprised of the stored page frame combined with a fast page offset portion derived from the segment base and the virtual address;

wherein the fast physical address is calculated prior to the generation of said calculated physical address.

62. The system of claim 61, wherein the fast physical address can be used for generating a fast memory access which is generated more quickly than a memory access resulting from said first physical address

63. The system of claim 61, including a cancellation circuit for cancelling the fast memory access if the fast physical address and first physical address are different.

64. The circuit of claim 61, wherein the fast physical address is generated prior to the generation of the first linear address.

65. The system of claim 61, wherein the stored page frame is generated in a prior address translation based on a prior virtual address.

66. A method of performing a translation of a virtual address in a computer system, said method including the steps of :

(a) calculating a fast physical address related to said virtual address; and

(b) calculating a linear address based on said virtual address; and

(c) calculating an actual physical address based on the linear address;

wherein step (a) is completed prior to the completion of step (c), and the fast physical address can be used to initiate a fast memory reference.

67. The method of claim 66, further including a step (d): cancelling the memory access if the fast physical address and actual physical address are different.

68. The method of claim 66, wherein the fast physical address is generated based on a combination of physical address information from a different virtual address, and partial linear address information relating to said virtual address.

69. The method of claim 66, wherein step (a) is completed prior to the completion of step (b).

70. A method of generating memory references based on virtual addresses in a computer system, said method including the steps of:

(a) generating tentative memory references based on said virtual addresses; and

(b) converting said virtual addresses to linear addresses during a segmentation operation; and

(c) converting said linear addresses to physical addresses during a paging operation, so that actual memory references can be made based on said physical addresses;

wherein the tentative memory reference can be generated while said virtual addresses are being converted in said first operation into said linear addresses.

71. The method of claim 70, further including a step (d): cancelling the tentative memory reference if the tentative memory reference and actual memory reference are different.

72. The method of claim 70, wherein the tentative memory reference is generated based on a combination of physical address information and partial linear address information relating to said virtual addresses.

73. The method of claim 70, wherein step (a) is completed prior to the completion of step (b).

74. A method of generating a fast memory reference using a fast physical address derived from a virtual address in a computer system, the method including the steps of:

(a) converting a portion of said virtual address into a partial linear address; and

(b) combining the partial linear address with physical address information obtained from a prior memory reference to generate said fast physical address;

(c) generating a memory reference based on the fast physical address;

(d) converting said virtual address into an actual physical address;

(e) cancelling the memory reference if the fast physical address and actual physical address are different.

75. The method of claim 74, wherein the fast physical address is generated prior to the generation of the linear address.

76. The method of claim 74, wherein the fast physical address is used to generate a fast memory access prior to the generation of the linear address.

77. A method of generating physical addresses from virtual addresses in a computer system, the method including the steps of:

(a) generating a first calculated linear address based on a first virtual address in a first operation; and

(b) generating a fast physical address in a second operation, the fast physical address including linear address information relating to said first virtual address and portions of physical address information relating to said first virtual address; and

(c) generating a first calculated physical address in a third operation based on the first calculated linear address;

wherein the fast physical address is generated prior to the generation of the first calculated physical address.

78. The method of claim 77, wherein the fast physical address is used to generate a tentative memory access prior to the generation of the first calculated physical address.

79. The method of claim 78, including a step (d): cancelling the tentative memory access if the fast physical address and first calculated physical address are different.

80. The method of claim 79, further including a step (e): generating a memory access request based on the first calculated physical address; and (f) storing physical address information relating to the first calculated physical address for use in a later address translation.

81. The method of claim 77, wherein the first and second operations overlap in time, and the fast physical address is generated prior to the generation of the first calculated linear address.

Remarks

Original claims 6 - 36 have been canceled. Claims 1-5 and 38 - 81 are presently pending.

Original claims 1-5 have been amended to specify that the linear address information and page frame information in the segment descriptor memory can be used to initiate a tentative memory reference, and without performing a virtual-linear-physical address translation. These features are neither disclosed nor suggested in Crawford '836, nor in any of the references submitted or considered to date. Since this feature is neither disclosed nor suggested by any of the art considered to date, applicant submits that the above claims are allowable at this time.

All of the new claims submitted herewith (38 - 81) are distinguishable over the art as well, since applicants believe they are the first to disclose and generally describe a *fast* physical address generator for use in a virtual-linear-physical address translation environment.

Enclosed also is an IDS for all the material references known to applicant. Most of these references were already considered in the parent application. For those references not considered, (1) EP 0-668-565 (Kranich); (2) U.S. Patent No. 4,400,774 (Toy); and (3) U.S. Patent No. 5,165,028, applicant submits the following explanation of why the present invention distinguishes thereover:

First, the Kranich reference does not make any mention of a combination of segmentation and paging based address translation, or a virtual-linear-physical conversion of any kind. In fact, it seems that a different type of address translation is described because at column 9, ll. 55 - column 10, ll. 35 it discusses a virtual address in the form of a virtual "page" system, which is not a "virtual address" of the segment identifier plus segment offset variety disclosed and claimed in the present invention. See present disclosure at p. 2, ll. 4-15 (discussing segmentation). In Kranich the virtual address is disclosed to already contain the page offset (see c.10, ll. 15-18) and therefore no segmentation (virtual to linear address conversion) operation is ever disclosed to occur. Hence there is no "segment," no "segment identifier," no "segmentation" unit or circuit, no "linear address," etc. etc., as those terms are used in the claims. Moreover, as indicated in the attached supplemental IDS, while Kranich claims a U.S. priority date prior to the present application (and may conceivably become 102(e) art at some point), the above European application was not published until after the filing date of the present invention, and therefore it is not believed to be prior art to the present application at this time. Since its status cannot be confirmed at this time, however, applicant requests that it be considered in the present application.

The Toy reference seems deficient for similar reasons. The Zulian reference also makes

mention of a logical (virtual) address that has an offset field that is the same as the physical address, see column 3, ll. 55-59, (like the Kranich virtual paging), and this disclosure is similarly different from the virtual addresses used in the present invention. The Toy reference construction of the virtual address also does not result in a virtual - linear - physical address conversion of the type generally described in the present disclosure.

In short, the present invention solves the timing bottleneck created by repetitive virtual-linear-physical address calculations in segmentation plus optional paging address translation systems (such as shown in Crawford '836) by using a tentative memory reference and this concept is neither disclosed nor suggested in any of the above references. Accordingly, the present claims are distinguishable over these newly submitted references as well, and should be allowable.

Should the Examiner believe it is necessary or fruitful to discuss any of the above points in person, Applicant is open to a teleconference (408-342-1862) at any convenient time.

Respectfully submitted,

Law+

Date: August 4, 1997

J. Nicholas Gross, Attorney, Reg. No. 34,175

I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to the Commissioner of Patents and Trademarks, this 4th of August, 1997.

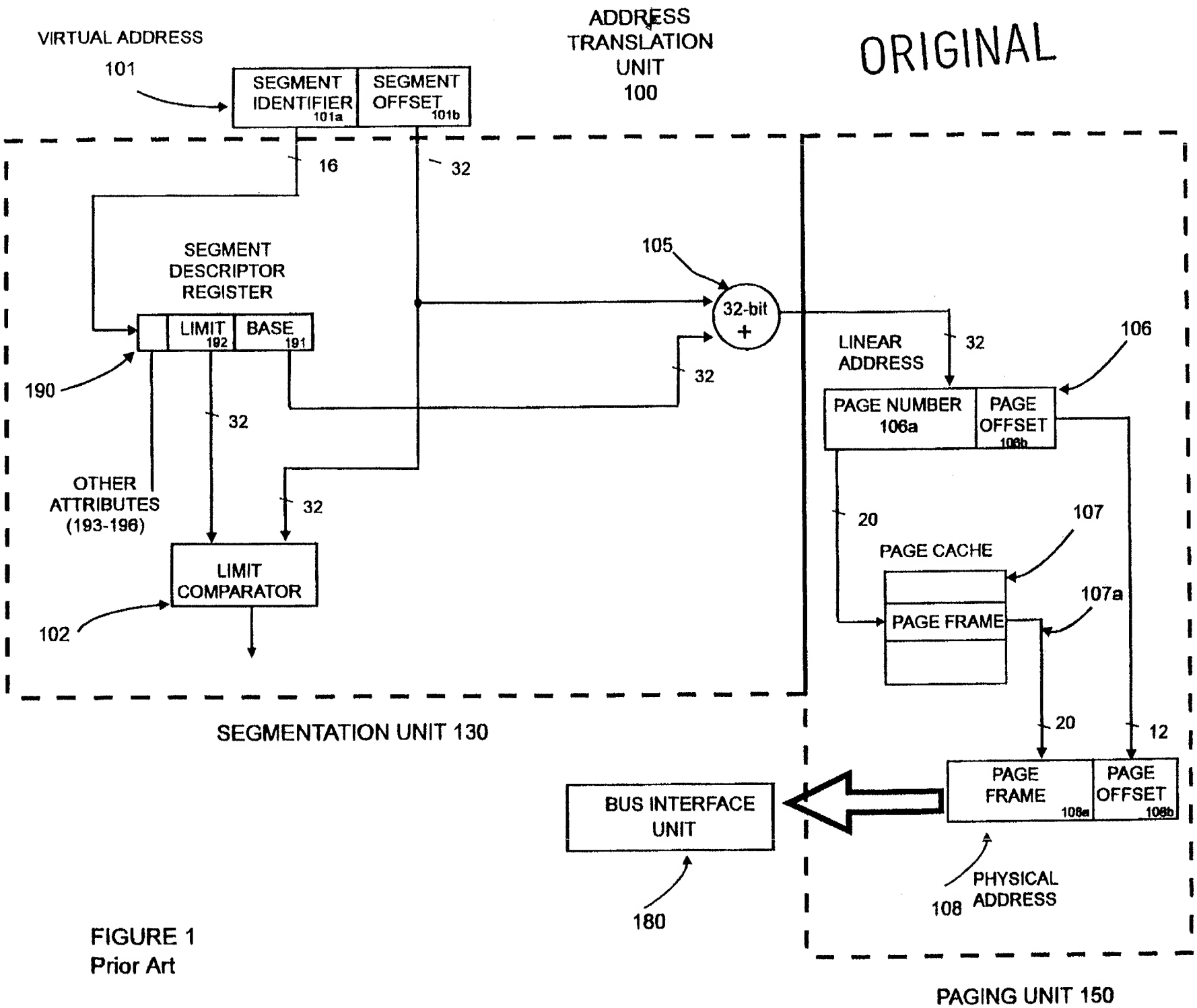
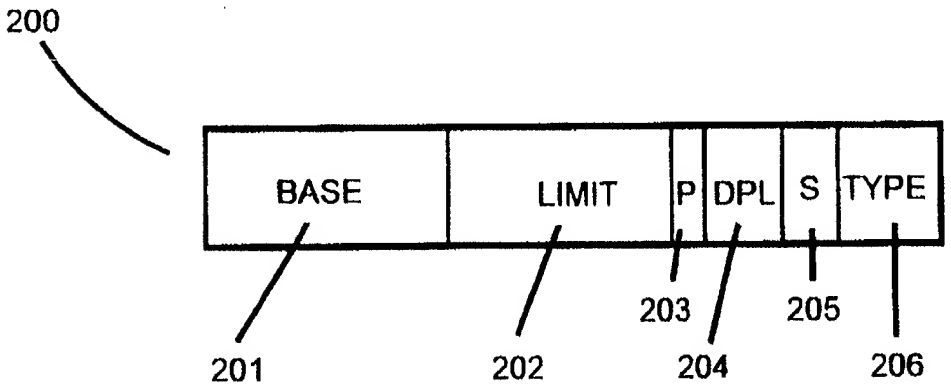


FIGURE 1
Prior Art

ORIGINAL



PRIOR ART SEGMENT DESCRIPTOR REGISTER

FIGURE 2A

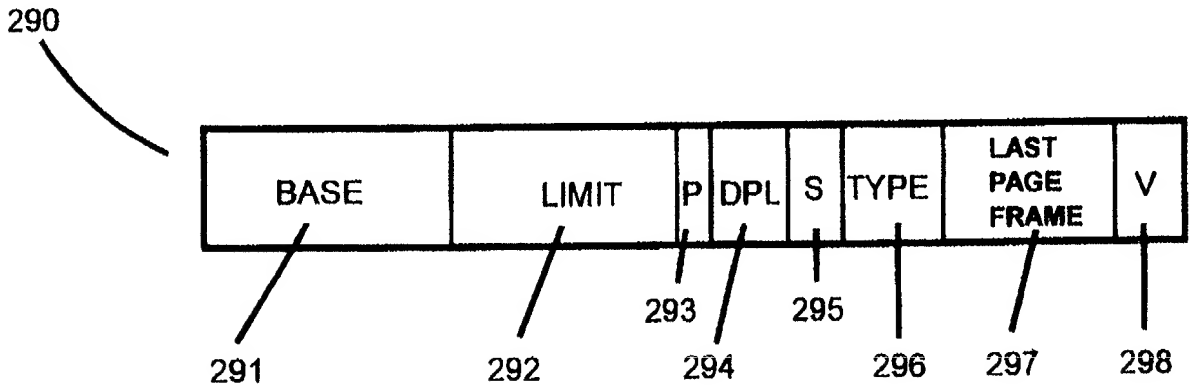


FIGURE 2B

SEGMENT DESCRIPTOR MEMORY

FIGURE 2

ORIGINAL

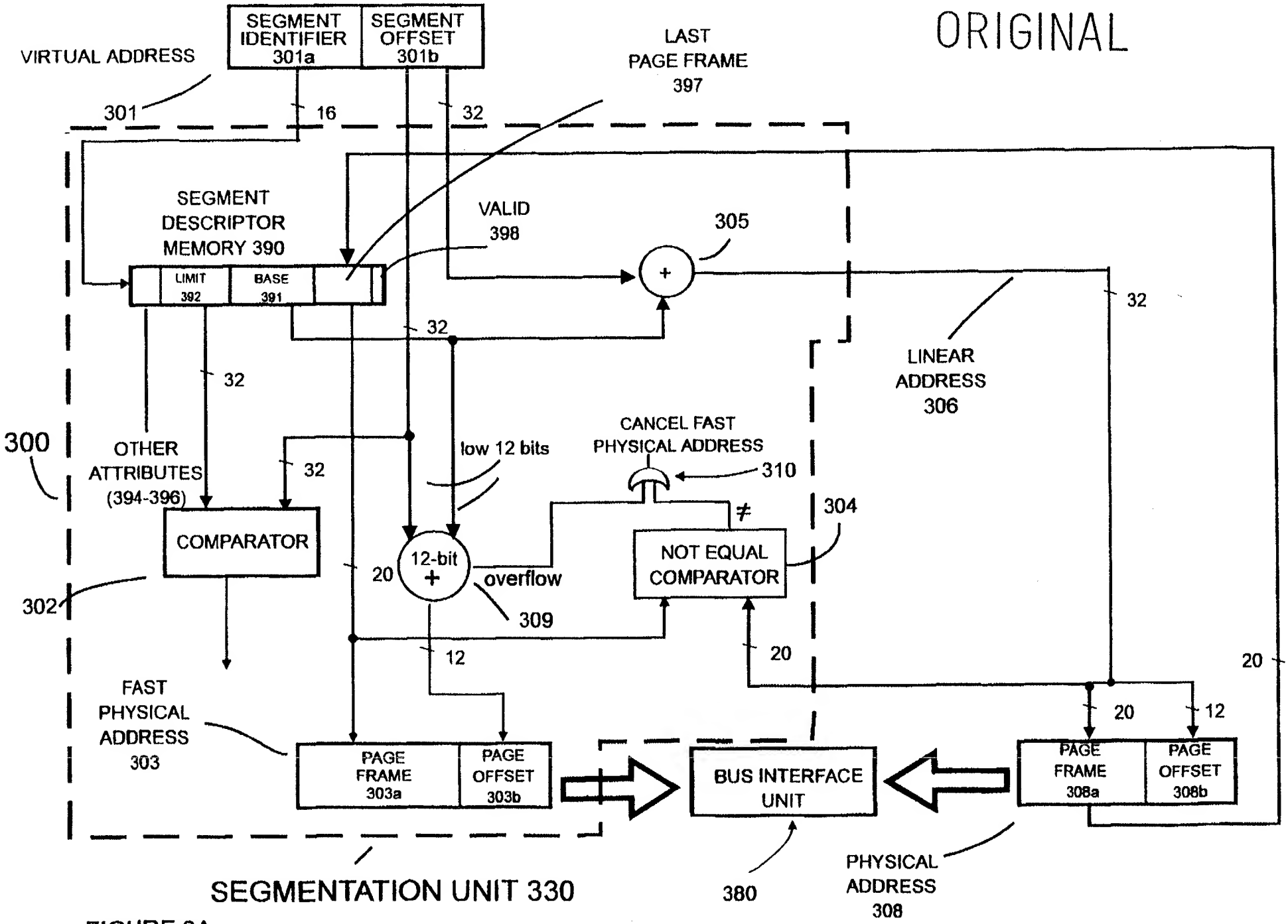


FIGURE 3A

ORIGINAL

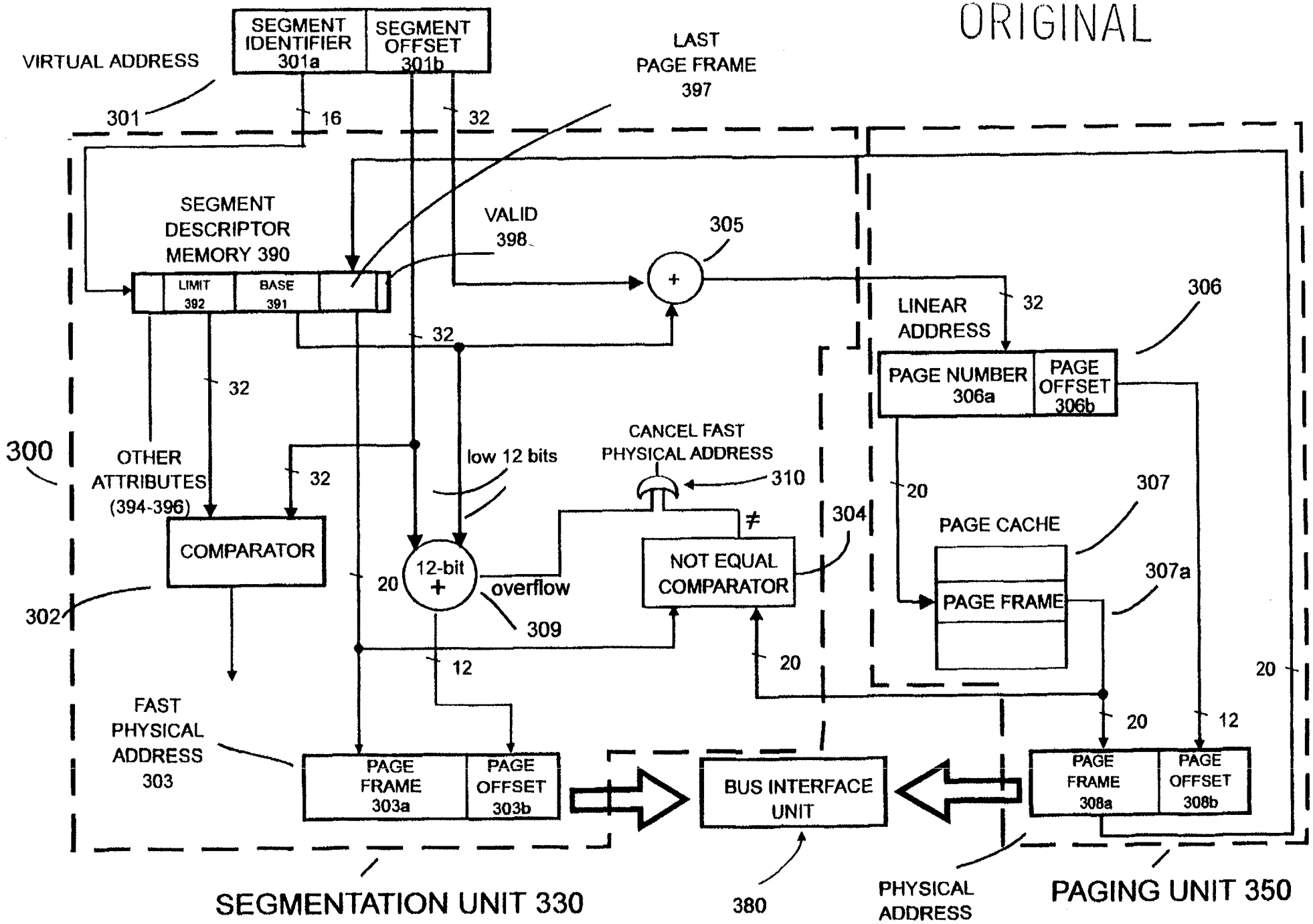


FIGURE 3B

ORIGINAL

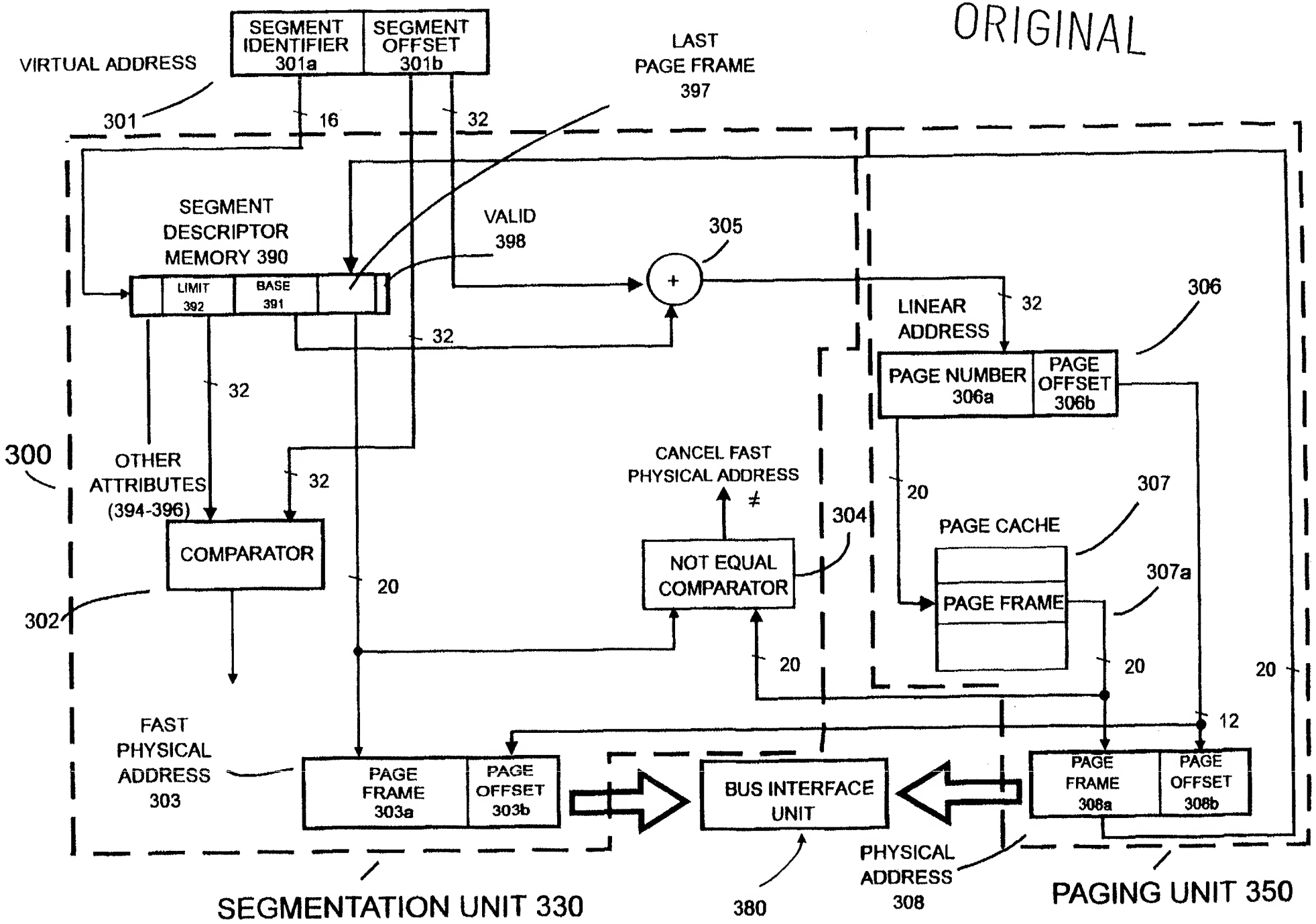


FIGURE 3C

Applicant or Patentee: *Richard A. Belgard* Attorney's Docket No: RAB 95-001
Serial or Patent No.: *Unknown*
Filed or Issued: *June 2, 1995*
For: *Improved Address Translation Mechanism and Method in a Computer System*

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9 (f) and 1.27 (b)) - INDEPENDENT INVENTOR

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9 (c) for purposes of paying reduced fees under section 41 (a) and (b) of Title 35, United States Code, to the Patent and Trademark described in

{X}the specification filed herewith
{ }application serial no.________, filed _________.
{ }patent no.____________, issued ___________.

I have not assigned, granted, conveyed or licensed and am under no obligation or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor under 37 CFR 1.9 (c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9 (d) or a nonprofit organization under 37 CFR 1.9 (e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

{X}no such person, concern, or organization
{ }persons, concerns, or organizations listed below*

*Note: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

FULL NAME *Richard A. Belgard*
ADDRESS *21250 Glenmont Drive Saratoga, CA 95070*

☒ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28 (b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Full name of inventor: *Richard A. Belgard*

Inventor's signature ______________________ Date 6/2/95